

-13-

REMARKS

The Examiner has rejected each of the claims under 35 U.S.C. 103(a) as being unpatentable over Jenevein et al. (U.S. Patent No.: 6,615,365). Applicant respectfully disagrees with such rejection, especially in view of the amendments made hereinabove to each of the independent claims. Specifically, applicant has at least substantially incorporated the subject matter of dependent Claims 3-4 et al. into each of the independent claims.

Specifically, in the latest action, the Examiner has relied on the following excerpts to meet applicant's claimed "creating a database of the known scanned regions of the verified file system; and validating an integrity of an object in the file system against the database of known scanned regions" (see this or similar, but not identical language in each of the independent claims).

"When the partition table 608 and/or the file system data 616 that would otherwise be used to locate an image 420 have been damaged, the image locator 620 can be used to determine where the image 420 was stored within the damaged partition 610. If the image 420 was not stored as a contiguous image, recovery will be facilitated if a FAT cluster chain or equivalent structure can be found (MFT runs in NTFS or inode information in UNIX-like file systems); if the image 420 was stored in a container then directory information will also be used. As noted, the cluster chain and directory information is normally stored in file system data, but this retrieval information may be alternatively or additionally stored inside the image 420 itself if compatibility with the existing Drive Image.RTM. format is not required. If compatibility is required, this retrieval information may be stored in the image container or the diagnostic and recovery partition. If an image cannot be found or recovered, because the media is irreparably damaged, because the user has deleted the image file(s) intentionally or inadvertently, or for other reasons, then an error is returned, the user is informed, and, in some implementations, the program exits." (col. 14, lines 49-67)

"In short, the system 600 saves necessary system data such as the partition table, boot record, root directory, and file allocation table (for FAT systems), Master File Table entries (for NTFS systems), boot block, super block, bitmap and inode information (for UNIX-like systems) or equivalent structures in other file systems. Thus, the system 600 is able to restore a desired image 420 when the partition table is damaged, when the boot record is damaged, when the file allocation table is damaged, when the

-14-

Master File Table is damaged, when the boot block, superblock, bitmap or inode information is damaged and when equivalent structures in other file systems are damaged. Sometimes an image cannot be found, because of damaged media or for other reasons, even using all of the backup procedures. In this case, an error is returned, the user is informed, and the program exits." (col. 15, lines 20-33)

"The image verifier 622 may also check the integrity of the contents of an image file by utilizing error checking techniques such as checksums, cyclic redundancy checks or other means known to the art. If errors or other exceptional conditions are detected by the image verifier 622 in any of its verifications, then appropriate measures are taken. If an error is discovered the verifier 622 may simply report the error, may attempt to fix the error by itself, or may attempt to use the image locator 620 and/or image restorer 624 to fix the error. In the case of a fatal error, conditions on the disk 606 that were changed by the implementing program are restored to the extent possible, a message may be passed to the end user (before or after the conditions are restored), and the implementing program is terminated." (col. 16, lines 5-20)

Applicant respectfully disagrees, as the above excerpts merely suggest verifying the integrity of contents of an image. There is absolutely no suggestion, however, of any sort of "validating an integrity of an object in the file system against the database of known scanned regions," as claimed. Only applicant teaches and claims the validation of an object in the file system against a database of known scanned regions, that is created as a result of scanning the file system (as claimed).

While the Examiner admits that "Jenevein does not teach specifically the uses of a database to contain information to verify the known scanned regions," he then argues that "Jenevein does teach of storing the information to verify the known scanned regions scanned regions in tables format for [a] software program to access for validation." Applicant respectfully disagrees, as it would not be obvious to provide a database of known scanned regions (which is absent from Jenevein), since there is no validation of an object in the file system against a database of known scanned regions. Since no such validation (of file system and/or portion thereof vs. a database of known scanned regions) exists in Jenevein, it would simply be *unobvious* to provide the non-existent claimed database of known scanned regions.

-15-

Again, only applicant teaches and claims the specific validation of an object in the file system with respect to a database of known scanned regions, that is created as a result of scanning the file system during a verification operation (as claimed).

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaack*, 947 F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991).

Applicant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art reference fails to teach or suggest all of the claim limitations, as noted above. Nevertheless, despite such paramount deficiencies and in the spirit of expediting the prosecution of the present application, applicant has at least substantially incorporated the subject matter of dependent Claims 3-4 et al. into each of the independent claims.

With respect to the subject matter of former Claim 4 (now at least substantially incorporated into each of the independent claims), the Examiner relies on the excerpts below to meet applicant's claimed technique "wherein the verifying comprises: receiving a file system event from a real-time monitoring system, the file system event indicating that an object in the file system has been accessed; and flagging the database of known scanned regions to indicate which of the known scanned regions was occupied by the accessed object."

"In some embodiments every block contains the following header information:

the file ID or unique image identifier which identifies which

-16-

file the block belongs to

the sequential ID which identifies each block's sequence number

the checksum which is used to verify the contents of the block,  
and

the image data." (col. 11, lines 8-15)

"Image 420 creation and image 420 location are closely related. For convenience, FIG. 6 shows an image locator 620 separate from the image creator 618, but the creation and location functions could be performed with overlapping or interwoven code in a given implementing program. The image locator 620 is used to locate one or more images 420 for data recovery, image updating, image deletion, image defragmentation, and similar operations pertinent to in-partition images. If multiple images 420 are found, the user can choose the image 420 desired, or the image 420 to operate on can be automatically chosen by creation date, name, or some other defining feature. For example, a partition 300 may contain both a factory image 424 and an end-user image 422. To restore data placed on the computer 600 after the purchase, the end-user image 422 would be chosen (unless it is incremental with respect to the factory image 424, in which case the factory image 424 would be used first and then the incremental end-user image 422 would be used)." (col. 14, lines 30-48)

"One way to implement the image locator 620 is to store portions of the system data in a known, fixed location within the imaged partition 300. The copied system data can be located, after the normal system data has been lost, by moving the disk head to the fixed location in question. This location would normally be marked as system, hidden, and read-only so it is not easily accessible to the end-user and is not easily deleted or overwritten. Another implementation stores the system data needed for image recovery outside the imaged partition 300 in a diagnostic and recovery partition 612. Yet another implementation, or a system that could also use one of the approaches already mentioned, backs up the necessary system data as recovery information onto a removable medium, such as a Zip drive, a Jaz drive, a WORM drive, a floppy (or floppies), a tape drive, and so on." (col. 15, lines 3-18)

After careful reviewing the above excerpts, and the remaining Jenevein reference for that matter, it is clear that Jenevein does not even "receiv[e] a file system event from a real-time monitoring system, the file system event indicating that an object in the file system has been accessed," let alone "flagging the database of known scanned regions to indicate which of the known scanned regions was occupied by the accessed object" (emphasis added), as claimed. Further, as admitted by the Examiner, Jenevein does not teach a database to contain information relating to the known scanned regions. To this

-17-

end, it would be *unobvious* to equip the non-existent database in Jenevein with the foregoing flagging functionality claimed by applicant.

Even still, in order to further distinguish Jenevein, now claimed in each of the independent claims is “validating [that] utilizes the flagging,” thus expanding upon the novel claimed validating operation. As mentioned above, Jenevein merely suggests the verification of the contents of an image, but fails to provide for validation of an object in the file system against a database of known scanned regions, that is created as a result of scanning the file system during a verification operation (as claimed). To this end, the use of the database flagging during the course of such validation provides for unique, improved operation.

Again, applicant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art reference fails to teach or suggest all of the claim limitations, as noted above. Thus, a notice of allowance or a specific prior art showing of all of applicant’s claim limitations, in combination with the remaining claim elements, is respectfully requested.

Applicant further notes that the prior art is also deficient with respect to the dependent claims. For example, with respect to Claim 9, the Examiner has relied on col. 11, lines 8-15; col. 13, line 15 – col. 14, line 27; and col. 15, lines 20-30 from the above reference to make a prior art showing of applicant’s claimed technique “wherein flagging comprises indicating which of the inodes and directory blocks were occupied by the accessed object” (see this or similar, but not necessarily identical language in each of the independent claims). In response, applicant notes that Jenevein fails to even teach the flagging (as noted above, as claimed), let alone indicating which inodes and directory blocks are occupied by the accessed object.

Thus, all of the independent claims are deemed allowable. Moreover, the remaining dependent claims are further deemed allowable, in view of their dependence on such independent claims.

-18-

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-1351 (Order No. NAI1P352/00.145.01).

Respectfully submitted,

Zilka-Kotab, PC.

  
Kevin J. Zilka  
Registration No. 41,429

P.O. Box 721120  
San Jose, CA 95172-1120  
408-505-5100